


(https://
profile.intra.42.fr)

SCALE FOR PROJECT INCEPTION (/PROJECTS/ INCEPTION)

You should evaluate 1 student in this team



Git repository

git@vogsphere.42antananarivo.mg:vogsphere/intra-uuid-4a3eeb43-84a9-4: 

Introduction


Please adhere to the following rules:

- Remain polite, courteous, respectful and constructive throughout the evaluation process. The well-being of the community depends on it.
- Identify the possible dysfunctions in the project of the student or group whose work is being evaluated. Take the time to discuss and debate the problems that may have been identified.
- You must consider that there might be some differences in how your peers might have understood the project's instructions and the scope of its functionalities. Always keep an open mind and grade them as honestly as possible. Pedagogy is useful only if peer evaluation is done seriously.

Guidelines

- Only grade the work submitted in the Git repository of the evaluated student or group.
- Double-check that the Git repository belongs to the student or students. Ensure that the project is the expected one. Also, check that 'git clone' is used in an empty directory.
- Check carefully that no malicious aliases were used to deceive you and make you evaluate something that is not the content of the official repository.
- To avoid any surprises, and if applicable, review together any scripts used to facilitate grading (such as testing or automation scripts).
- If you have not completed the assignment you are going to evaluate, you must read the entire subject before starting the evaluation process.
- Use the available flags to report an empty repository, a non-functioning program, a Norm error, cheating, etc.
In these cases, the evaluation process ends, and the final grade is 0, or -42 in the case of cheating. However, except in cases of cheating, students are strongly encouraged to review the submitted work together to identify any mistakes that should not be repeated in the future.

Attachments

 subject.pdf (<https://cdn.intra.42.fr/pdf/pdf/189602/en.subject.pdf>)

Preliminaries

If cheating is suspected, the evaluation stops here. Use the "Cheat" flag to report this. Make this decision calmly and wisely, and please, use this button with caution.

Preliminary tests

- For example, the use of a local .env file to store info is allowed, and/or also the use of docker secrets In case any credentials, API keys, or passwords are available in the git repository and outside of secrets files created during the evaluation, the evaluation stops and the mark is 0.
- Defense can only happen if the evaluated learner or group is present. This way everybody learns by sharing knowledge with each other.
- If no work has been submitted (or wrong files, wrong directory, or wrong filenames), the grade is 0, and the evaluation process ends.
- For this project, you have to clone their Git repository on their station.

 Yes

 No

General instructions

General instructions

- For the entire evaluation process, if you don't know how to check a requirement, or verify anything, the evaluated learner has to help you.
- Ensure that all the files required to configure the application are located inside a srcs folder. The srcs folder must be located at the root of the repository.
- Ensure that a Makefile is located at the root of the repository.
- Before starting the evaluation, run this command in the terminal: "docker stop \$(docker ps -qa); docker rm \$(docker ps -qa); docker rmi -f \$(docker images -qa); docker volume rm \$(docker volume ls -q); docker network rm \$(docker network ls -q) 2>/dev/null"
- Read the docker-compose.yml file. There mustn't be 'network: host' in it or 'links:'. Otherwise, the evaluation ends now.
- Read the docker-compose.yml file. There must be 'network(s)' in it. Otherwise, the evaluation ends now.
- Examine the Makefile and all the scripts in which Docker is used. There mustn't be '--link' in any of them. Otherwise, the evaluation ends now.
- Examine the Dockerfiles. If you see 'tail -f' or any command run in background in any of them in the ENTRYPOINT section, the evaluation ends now. Same thing if 'bash' or 'sh' are used but not for running a script (e.g. 'nginx & bash' or 'bash').
- Examine the Dockerfiles. The containers must be built either from the penultimate stable version of Alpine or Debian.
- If the entrypoint is a script (e.g., ENTRYPOINT ["sh", "my_entrypoint.sh"], ENTRYPOINT ["bash", "my_entrypoint.sh"]), ensure it runs no program in background (e.g. 'nginx & bash').
- Examine all the scripts in the repository. Ensure none of them runs an infinite loop. The following are a few examples of prohibited commands: 'sleep infinity', 'tail -f /dev/null', 'tail -f /dev/random'
- Run the Makefile.

 Yes

 No

Mandatory part

This project involves setting up a small infrastructure composed of different services using docker compose. Ensure

that all of the following points are correct.

Activity overview

The evaluated learner has to explain to you in simple terms:

- How Docker and docker compose work
- The difference between a Docker image used with docker compose and without docker compose
- The benefit of Docker compared to VMs
- The pertinence of the directory structure required for this project (an example is provided in the subject's PDF file).

✓ Yes

✗ No

README check

- Ensure that a README.md file is present at the root of the repository.
- The very first line must follow the required format: "This project has been created as part of the 42 curriculum by <login...>" (italicized).
- Check that the README contains at least the required sections: Description, Instructions, Resources (with explanation of how AI was used).
- If any of these elements are missing, the evaluation ends now.

✓ Yes

✗ No

Documentation check

- Ensure that both USER_DOC.md and DEV_DOC.md files are present at the root of the repository.
- USER_DOC.md must provide basic usage instructions for an end user or administrator (start/stop the stack, access the website and admin panel, manage credentials, basic checks).
- DEV_DOC.md must provide developer-oriented instructions (prerequisites, setup, Makefile usage, docker compose commands, data persistence).
- If any of these files are missing or empty, the review ends now.

✓ Yes

✗ No

Simple setup

- Ensure that NGINX can be accessed by port 443 only. Once done, open the page.
- Ensure that a SSL/TLS certificate is used.
- Ensure that the WordPress website is properly installed and configured (you shouldn't see the WordPress Installation page). To access it, open <https://login.42.fr> (<https://login.42.fr>) in your browser, where login is the login of the evaluated learner. You shouldn't be able to access the site via <http://login.42.fr> (<http://login.42.fr>). If something doesn't work as expected, the evaluation process ends now.

✓ Yes

✗ No

Docker Basics

- Start by checking the Dockerfiles. There must be one Dockerfile per service. Ensure that the Dockerfiles are not empty files. If it's not the case or if a Dockerfile is missing, the evaluation process ends now.
- Make sure the evaluated learner has written their own Dockerfiles and built their own Docker images. Indeed, it is forbidden to use ready-made ones or to use services such as DockerHub.
- Ensure that every container is built from the penultimate stable version of Alpine/Debian. If a Dockerfile does not start with 'FROM alpine:X.X.X' or 'FROM debian:XXXXX', or any other local image, the evaluation process ends now.
- The Docker images must have the same name as their corresponding service. Otherwise, the evaluation process ends now.
- Ensure that the Makefile has set up all the services via docker compose. This means that the containers must

have been built using docker compose and that no crash happened. Otherwise, the evaluation process ends.

☒ Yes

☐ No

Docker Network

- Ensure that docker-network is used by checking the docker-compose.yml file. Then run the 'docker network ls' command to verify that a network is visible.
- The evaluated learner has to give you a simple explanation of docker-network.
- If any of the above points is not correct, the evaluation process ends now.

☒ Yes☐ No

NGINX with SSL/TLS

- Ensure that there is a Dockerfile.
- Using the 'docker compose ps' command, ensure that the container was created (using the flag '-p' is authorized if necessary).
- Try to access the service via http (port 80) and verify that you cannot connect.
- Open <https://login.42.fr/> (<https://login.42.fr/>) in your browser, where login is the login of the evaluated learner. The displayed page must be the configured WordPress website (you shouldn't see the WordPress Installation page).
- The use of a TLS v1.2 or TLS v1.3 certificate is mandatory and must be demonstrated. The SSL/TLS certificate doesn't have to be recognized. A self-signed certificate warning may appear.
- If any of the above points is not clearly explained and correct, the evaluation process ends now.

☒ Yes☐ No

WordPress with php-fpm and its volume

- Ensure that there is a Dockerfile.
- Ensure that there is no NGINX in the Dockerfile.
- Using the 'docker compose ps' command, ensure that the container was created (using the flag '-p' is authorized if necessary).
- Ensure that there is a Volume. To do so: Run the command 'docker volume ls' then 'docker volume inspect <volume name>'. Verify that the result in the standard output contains the path '/home/login/data/', where login is the login of the evaluated learner.
- Ensure that you can add a comment using the available WordPress user.
- Sign in with the administrator account to access the Administration dashboard. The Admin username must not include 'admin' or 'Admin' (e.g., admin, administrator, Admin-login, admin-123, and so forth).
- From the Administration dashboard, edit a page. Verify on the website that the page has been updated.
- If any of the above points is not correct, the evaluation process ends now.

☒ Yes☐ No

MariaDB and its volume

- Ensure that there is a Dockerfile.
- Ensure that there is no NGINX in the Dockerfile.
- Using the 'docker compose ps' command, ensure that the container was created (using the flag '-p' is authorized if necessary).
- Ensure that there is a Volume. To do so: Run the command 'docker volume ls' then 'docker volume inspect <volume name>'. Verify that the result in the standard output contains the path '/home/login/data/', where login is the login of the evaluated learner.
- The evaluated learner must be able to explain you how to login into the database. Verify that the database is not empty.
- If any of the above points is not correct, the evaluation process ends now.

☒ Yes☐ No

Persistence!

- This part is pretty straightforward. You have to reboot the virtual machine. Once it has restarted, launch 'docker compose up' again. Then, verify that everything is functional, and that both WordPress and MariaDB are configured. The changes you made previously to the WordPress website should still be here. If any of the above points is not correct, the evaluation process ends now.

✓ Yes

✗ No

Configuration modification

- During the defense, the reviewer must ask the evaluated person to modify the configuration of one service (for example by changing the port it is using).
- The reviewer is free to choose which service and which new port, as long as the port is available on the system.
- After the change, the evaluated person must rebuild and restart the project.
- The service must remain accessible and functional with the new configuration.
- If the modification cannot be performed or the service no longer works, the evaluation ends now.

✓ Yes

✗ No

Bonus

For this project, the bonus part is intended to be simple. A Dockerfile must be written for each additional service. Thus, each service will run inside its own container and will have, if necessary, its dedicated volume. Evaluate the bonus part only if the mandatory part has been completed entirely and perfectly. Perfect means the mandatory part has been fully completed and functions without any malfunctions. If you have not passed ALL the mandatory requirements, your bonus part will not be evaluated at all.

Bonus

Bonus list:

- Set up redis cache for your WordPress website in order to properly manage the cache.
- Set up a FTP server container pointing to the volume of your WordPress website.
- Create a simple static website in the language of your choice except PHP (yes, PHP is excluded). For example, a showcase site or a site for presenting your resume.
- Set up Adminer.
- Set up a service of your choice that you think is useful. During the defense, you will have to justify your choice. Add 1 point per bonus authorized in the subject. Verify and test the proper functioning and implementation of each additional service. For the free choice service, the evaluated student must provide a simple explanation of how it works and why they believe it is useful.

Rate it from 0 (failed) through 5 (excellent)

**Ratings**

Don't forget to check the flag corresponding to the defense

✓ Ok

★ Outstanding project

Empty work

📄 Incomplete work

📄 Cheat

💥 Crash

⚠ Concerning situation

💬 Can't support / explain code

Conclusion

Leave a comment on this evaluation (2048 chars max)



Finish evaluation

API General
Terms of Use
([https://
profile.intra.42.fr/
legal/terms/33](https://profile.intra.42.fr/legal/terms/33))

Declaration on
the use of
cookies ([https://
profile.intra.42.fr/
legal/terms/2](https://profile.intra.42.fr/legal/terms/2))

Privacy policy
([https://
profile.intra.42.fr/
legal/terms/5](https://profile.intra.42.fr/legal/terms/5))

General term of
use of the site
([https://
profile.intra.42.fr/
legal/terms/6](https://profile.intra.42.fr/legal/terms/6))

Règlement
intérieur ([https://
profile.intra.42.fr/
legal/terms/10](https://profile.intra.42.fr/legal/terms/10))

Terms of use for
video
surveillance
([https://
profile.intra.42.fr/
legal/terms/14](https://profile.intra.42.fr/legal/terms/14))

Legal notices
([https://
profile.intra.42.fr/
legal/terms/3](https://profile.intra.42.fr/legal/terms/3))